

# A Semidefinite Relaxation for Air Traffic Flow Scheduling

Alexandre d'Aspremont\* Laurent El Ghaoui†

February 1, 2008

## Abstract

We first formulate the problem of optimally scheduling air traffic flow with sector capacity constraints as a mixed integer linear program. We then use semidefinite relaxation techniques to form a convex relaxation of that problem. Finally, we present a randomization algorithm to further improve the quality of the solution. Because of the specific structure of the air traffic flow problem, the relaxation has a single semidefinite constraint of size  $dn$  where  $d$  is the maximum delay and  $n$  the number of flights.

## 1 Introduction

In this paper, given a schedule of flights and their routes we solve the problem of finding a new schedule that satisfies a list of sector capacity constraints and minimizes the total delay compared to the original scheduled. While the optimal routing problem with weather uncertainty and capacity constraints is essentially intractable, [NEG05] show that robust optimal routing of a single aircraft under weather and traffic uncertainty can be solved efficiently as a robust Markov dynamic programming (MDP) problem. So the problem we solve here should be seen as a second phase: given the routes computed using MDP, our relaxation produces an optimal schedule, satisfying capacity constraints while minimizing the total delay. While this scheduling problem is significantly simpler than the global routing problem, it still combinatorial (and in fact NP-Hard), hence we need to formulate a relaxation to obtain approximate solutions efficiently. We can interpret this problem as particular case of job shop scheduling problem where a task corresponds to an aircraft occupying a sector at a certain time, with the aircraft routes correspond to lists of tasks that have to be performed in sequence and where the sector capacities correspond to server capacities. In our case, a few key distinctions simplify the problem formulation. First, the different tasks are independent (one aircraft's route is not dependent on another). Second, for each aircraft, no time gap is allowed between two tasks (we assume that aircraft can't be held en-route).

Scheduling problems are notoriously hard combinatorial problems. Classic instances include the job-shop scheduling problem (see [CB76] for example) or the travelling salesman problem (see [RSL77] for example). Semidefinite relaxations and randomization techniques have had excellent track record, which can be traced to [LS91], [Ali95], [GW95] and [PW95] among others. Recently, [FMOF01] applied these techniques to aircraft conflict avoidance for free flight, by adjusting aircraft speed and bearing.

In this paper, we first formulate the air traffic flow scheduling problem as a mixed integer linear programming problem. We then apply the lifting procedure of [GW95] to formulate a semidefinite relaxation

---

\*ORFE Department, Princeton University, Princeton, NJ 08544. alexandre.daspremont@m4x.org

†EECS Department, U.C. Berkeley, Berkeley, CA 94720. elghaoui@eecs.berkeley.edu

of the problem. Because of the structure of our problem here, we only have  $dn$  binary variables, where  $d$  is the maximum delay and  $n$  the number of flights, which means that the semidefinite relaxation has one semidefinite constraint of size  $dn + 1$ , allowing to scale better than classic scheduling problem relaxations. We then detail a randomization procedure based on that of [GW95] that uses the matrix solution of the semidefinite relaxation to further improve the quality of the solution. While [FMOF01] focus on free flight and conflict avoidance, our main concern is on meeting sector capacity constraints, a key limiting factor in the European airspace, by adjusting delay at departure. This relative simplicity allows the algorithm to scale very well with the number of aircraft. Finally, because of its particular structure, we show that this relaxation is amenable to first-order methods for large-scale semidefinite optimization such as those detailed in [HR00] and [Nes04] which are natural algorithms for solving large-scale problems for which a low precision is required.

The paper is organized as follows. The second section defines the main scheduling problem and formulates it as a mixed integer linear program. The third section derives the semidefinite relaxation of that problem. In the fourth section, we show how to exploit the result of this relaxation to further improve the solution using a randomization technique. Finally, in section five, we present some numerical results.

## 2 Problem formulation

Suppose we are given routes for  $n$  aircraft flying across an airspace composed of  $m$  sectors with capacities given by  $C \in \mathbf{R}^m$ . We decompose a particular day into  $T$  periods, so that a particular flight route starting at time  $s$  can be represented by a matrix  $R^{(i,s)} \in \mathbf{R}^{m \times T}$  such that:

$$\begin{cases} R_{jt}^{(i,s)} = 1 & \text{if aircraft } i \text{ is in sector } j \text{ at time } t \\ R_{jt}^{(i,s)} = 0 & \text{if not.} \end{cases}$$

We can formulate the problem of minimizing total delay while satisfying capacity constraints as:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n \sum_{j=0}^d x_{ij} j \\ \text{subject to} & \sum_{i=1}^n \sum_{j=0}^d x_{ij} R^{(i,s+j)} \leq C \\ & \sum_{j=0}^d x_{ij} = 1 \\ & x_{ij} \in \{0, 1\}, \quad i = 1, \dots, n, \quad j = 0, \dots, d, \end{array} \tag{1}$$

in the variable  $x_{ij} \in \mathbf{R}^{n \times (d+1)}$  where  $d$  is the maximum delay (in units of time). Here,  $x_{ij} = 1$  means that aircraft  $i$  will be delayed by  $j$  units of time (because  $x$  is a binary variable and  $\sum_{j=0}^d x_{ij} = 1$ , there can only be a single nonzero  $x_{ij}$  for each aircraft  $i$ ). The first constraint makes sure that sector capacity constraints are met and the objective is the sum of aircraft delays.

## 3 Semidefinite Relaxation

In this section, we apply the *lifting* procedure detailed in [GW95] to form a semidefinite relaxation of problem (1). We can rewrite (1) as a non-convex quadratic program (QP):

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n \sum_{j=0}^d x_{ij} j \\ \text{subject to} & \sum_{i=1}^n \sum_{j=0}^d x_{ij} R^{(i,s+j)} \leq C \\ & \sum_{j=0}^d x_{ij} = 1 \\ & x_{ij}^2 - x_{ij} = 0, \quad i = 1, \dots, n, \quad j = 0, \dots, d, \end{array} \tag{2}$$

in the variable  $x_{ij} \in \mathbf{R}^{n \times (d+1)}$ . We can rewrite this problem as:

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^n \sum_{j=0}^d x_{ij} j \\ & \text{subject to} && \sum_{i=1}^n \sum_{j=0}^d x_{ij} R^{(i,s+j)} \leq C \\ & && \sum_{j=0}^d x_{ij} = 1, \quad i = 1, \dots, n \\ & && X = \text{vec}(x) \text{vec}(x)^T \\ & && \text{diag}(X) = \text{vec}(x) \end{aligned} \tag{3}$$

in the variables  $x_{ij} \in \mathbf{R}^{n \times (d+1)}$  and  $X \in \mathbf{S}^{n(d+1)}$ . This is a non-convex quadratic program and is computationally hard. We now detail how to obtain a convex relaxation of problem (3) using Lagrangian duality.

### 3.1 Lagrangian relaxation

Let us start from a general nonconvex quadratically constrained quadratic program (QCQP):

$$\begin{aligned} & \text{minimize} && x^T P_0 x + q_0^T x + r_0 \\ & \text{subject to} && x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \dots, m, \end{aligned} \tag{4}$$

with variable  $x \in \mathbf{R}^n$ , and parameters  $P_i \in \mathbf{S}^n$ ,  $q_i \in \mathbf{R}^n$ , and  $r_i \in \mathbf{R}$ . We form the Lagrangian,

$$L(x, \lambda) = x^T \left( P_0 + \sum_{i=1}^m \lambda_i P_i \right) x + \left( q_0 + \sum_{i=1}^m \lambda_i q_i \right)^T x + r_0 + \sum_{i=1}^m \lambda_i r_i.$$

To find the dual function, we minimize over  $x$ , using the general formula (see example 4.5 in [BV04]):

$$\inf_{x \in \mathbf{R}^n} x^T P x + q^T x + r = \begin{cases} r - \frac{1}{4} q^T P^\dagger q, & \text{if } P \succeq 0 \text{ and } q \in \mathcal{R}(P) \\ -\infty, & \text{otherwise.} \end{cases}$$

The dual function is then:

$$\begin{aligned} g(\lambda) &= \inf_{x \in \mathbf{R}^n} L(x, \lambda) \\ &= -\frac{1}{4} \left( q_0 + \sum_{i=1}^m \lambda_i q_i \right)^T \left( P_0 + \sum_{i=1}^m \lambda_i P_i \right)^\dagger \left( q_0 + \sum_{i=1}^m \lambda_i q_i \right) + \sum_{i=1}^m \lambda_i r_i + r_0. \end{aligned}$$

We can form the dual of (4), using Schur complements (cf. §A.5.5):

$$\begin{aligned} & \text{maximize} && \gamma + \sum_{i=1}^m \lambda_i r_i + r_0 \\ & \text{subject to} && \begin{bmatrix} (P_0 + \sum_{i=1}^m \lambda_i P_i) & (q_0 + \sum_{i=1}^m \lambda_i q_i) / 2 \\ (q_0 + \sum_{i=1}^m \lambda_i q_i)^T / 2 & -\gamma \end{bmatrix} \succeq 0 \\ & && \lambda_i \geq 0, \quad i = 1, \dots, m, \end{aligned} \tag{5}$$

in the variable  $\lambda \in \mathbf{R}^m$ . As the dual to (4), this is a convex program, it is in fact a semidefinite program (SDP). This SDP is called the *Lagrangian relaxation* of the nonconvex QCQP. It can be solved efficiently and gives a lower bound on the optimal value of the nonconvex QCQP. We form take the dual of program (5) (see [BV04, §5.9.2]):

$$\begin{aligned} & \text{minimize} && \text{Tr}(XP_0) + q_0^T x + r_0 \\ & \text{subject to} && \text{Tr}(XP_i) + q_i^T x + r_i \leq 0, \quad i = 1, \dots, m, \\ & && \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0, \end{aligned} \tag{6}$$

with variable  $x \in \mathbf{R}^n$ ,  $X \in \mathbf{S}^n$  and parameters  $P_i \in \mathbf{S}^n$ ,  $q_i \in \mathbf{R}^n$ , and  $r_i \in \mathbf{R}$ .

### 3.2 A relaxation of the scheduling problem

Using the results of the previous section, we can form the Lagrangian relaxation of (3) as:

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^n \sum_{j=0}^d x_{ij} j \\ & \text{subject to} && \sum_{i=1}^n \sum_{j=0}^d x_{ij} R^{(i,s+j)} \leq C \\ & && \sum_{j=0}^d x_{ij} = 1, \quad i = 1, \dots, n, \\ & && \begin{bmatrix} X & \text{vec}(x) \\ \text{vec}(x)^T & 1 \end{bmatrix} \succeq 0 \\ & && \text{diag}(X) = \text{vec}(x), \end{aligned} \tag{7}$$

which is a semidefinite program in the variables  $x_{ij} \in \mathbf{R}^{n \times (d+1)}$  and  $X \in \mathbf{S}^{n(d+1)}$  and can be solved efficiently. The objective of this program is a *lower bound* on the global solution.

### 3.3 First-order methods

An important structural property of problem (7) is that, because  $\text{diag}(X) = \text{vec}(x)$  and  $\mathbf{1}^T x = n$  the matrix variable:

$$\begin{bmatrix} X & \text{vec}(x) \\ \text{vec}(x)^T & 1 \end{bmatrix}$$

has constant trace equal to  $n + 1$ . This means that the dual of (7) is a maximum conic eigenvalue minimization problem for which efficient first-order methods such as the spectral bundle algorithm of [HR00] and the optimal first-order method of [Nes04]. In Section 5, we present numerical experiments using the SBmethod code by [HR00] on increasingly large randomly chosen problems. SBmethod solves large-scale dense instances for which it returns a solution  $(x, X)$ . It can also solve much larger problems by exploiting sparsity, in which case however it only returns the optimal  $x$ , which somewhat decreases the performance of the randomization methods detailed in the next section.

## 4 Randomization

The Lagrangian relaxation techniques developed in §3.1 provided lower bounds on the optimal value of the program in (4), but did not however give any particular hint on how to compute good feasible points. The semidefinite relaxation in (6) produces a positive semidefinite or covariance matrix together with the lower bound on the objective. In this section, we exploit this additional output to compute good approximate solutions with, in some cases, hard bounds on their suboptimality.

### 4.1 Randomization

In section 3.1, the original nonconvex QCQP:

$$\begin{aligned} & \text{minimize} && x^T P_0 x + q_0^T x + r_0 \\ & \text{subject to} && x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \dots, m, \end{aligned}$$

was relaxed into:

$$\begin{aligned} & \text{minimize} && \text{Tr}(X P_0) + q_0^T x + r_0 \\ & \text{subject to} && \text{Tr}(X P_i) + q_i^T x + r_i \leq 0, \quad i = 1, \dots, m, \\ & && \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0. \end{aligned} \tag{8}$$

The last (Schur complement) constraint being equivalent to  $X - xx^T \succeq 0$ , if we suppose  $x$  and  $X$  are the solution to the relaxed program in (8), then  $X - xx^T$  is a covariance matrix. If we pick  $x$  as a Gaussian variable with  $x \sim \mathcal{N}(x, X - xx^T)$ ,  $x$  will solve the nonconvex QCQP in (4) “on average” over this distribution, meaning:

$$\begin{aligned} & \text{minimize} && \mathbf{E}(x^T P_0 x + q_0^T x + r_0) \\ & \text{subject to} && \mathbf{E}(x^T P_i x + q_i^T x + r_i) \leq 0, \quad i = 1, \dots, m, \end{aligned}$$

and a “good” feasible point can then be obtained by sampling  $x$  a sufficient number of times, then simply keeping the best feasible point. Of course the direct sampling technique above does not guarantee that a feasible point will be found. In particular, if the program includes an equality constraint, then this method will certainly fail. However, it is sometimes possible to directly project the random samples onto the feasible set. As we will see below, this is the case here

## 4.2 Randomized schedules

Suppose we have solved

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^n \sum_{j=0}^d x_{ij} j \\ & \text{subject to} && \sum_{i=1}^n \sum_{j=0}^d x_{ij} R^{(i,s+j)} \leq C \\ & && \sum_{j=0}^d x_{ij} = 1, \quad i = 1, \dots, n, \\ & && \begin{bmatrix} X & \mathbf{vec}(x) \\ \mathbf{vec}(x)^T & 1 \end{bmatrix} \succeq 0 \\ & && \mathbf{diag}(X) = \mathbf{vec}(x), \end{aligned}$$

to get an optimal  $x_{ij} \in \mathbf{R}^{n \times (d+1)}$  and  $X \in \mathbf{S}^{n \times (d+1)}$ . We can simply sample a Gaussian variable  $u \sim \mathcal{N}(x, X - xx^T)$  and compute its projection  $v$  on

$$\{0, 1\}^{n \times (d+1)} \cap \left\{ x \in \mathbf{R}^{n \times (d+1)} : \sum_{j=0}^d x_{ij} = 1, i = 1, \dots, n \right\}.$$

We then compute the delay for each feasible random schedule  $v$  and keep the best solution.

## 5 Numerical Example

To fix ideas, let us begin by solving a very simple scheduling example. Suppose that there are only four sectors and two flights.



Each sector has capacity one. In our format, a solution where both flights leave on time will be represented by  $x = (1, 0, 1, 0)$  and a solution where flight 1 is delayed by one unit of time will be written as  $x = (0, 1, 1, 0)$ . Flight one starts from sector 1 and ends in sector 4, while flight 2 starts from sector 3 and also ends in sector 4. It takes each flight one unit of time to cross each sector and both flights are scheduled to

depart at time 1 and conflict in sectors 2 and 3. The maximum delay is 1 unit of time. The problem variables are  $x \in \mathbf{R}^4$  and  $X \in \mathbf{S}^4$ , the linear sector capacity constraints impose:

$$\left( \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right) x \leq \mathbf{1}$$

and the objective vector is given by  $c = (0, 1, 0, 1)$ . We solve (7) using SEDUMI by [Stu99] to get the following solution:

$$x = \begin{pmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{pmatrix} \quad \text{and} \quad X = \begin{pmatrix} 0.50 & 0.24 & 0.24 & 0.24 \\ 0.24 & 0.50 & 0.24 & 0.24 \\ 0.24 & 0.24 & 0.50 & 0.24 \\ 0.24 & 0.24 & 0.24 & 0.50 \end{pmatrix}.$$

Here, the solution is not binary valued. The reason for this is symmetry, the problem too simple and symmetric in flights one and two, so the solution is a mix of the two optimal solutions. This is easily fixed by adding a small random perturbation to the objective (implicitly breaking the tie by giving one aircraft priority over another). The solution is then:

$$x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad \text{and} \quad X = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

Here, the fact that  $X = xx^T$  also shows that the relaxation is tight and that the solution  $x$  is globally optimal.

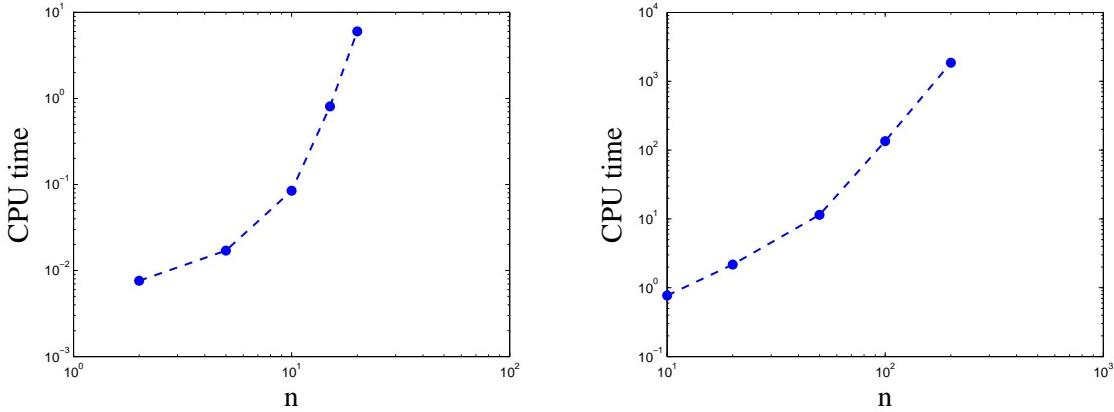
## 5.1 Computing times

Here, we generate random problems in an airspace with 50 sectors and a maximum delay of 2 units of time. In Figure 1, plot of CPU time (in minutes) versus number of aircraft for these problems.

With  $n = 20$  for example, the computing time is about 6 minutes, for a problem with  $10^{18}$  possible schedules. While general-purpose interior point solvers such as SEDUMI by [Stu99] can solve reasonably large problem instances. Using the fact that the semidefinite program in (7) has constant trace, more specialized first order methods (see [HR00] or [dEGJL05] and [d'A05] for example) can be used to solve much larger problems.

## 5.2 Randomization

To illustrate the randomization method detailed in §4, we first solve problem (7) on a random problem with 5 sectors, 3 flights and a maximum delay of 4 units of time. The solution  $x$  to (7) has an objective

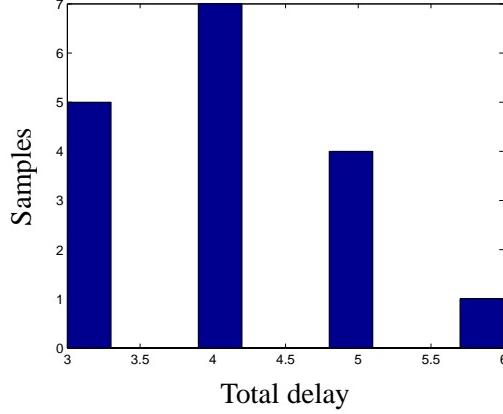


**Figure 1:** Log-log plot of CPU time (in minutes) versus number of aircraft for randomly generated problems, *left*: interior point methods, *right*: spectral bundle method.

value of 3, which gives a lower bound on the global optimum. We then generate 100 sample points from  $u \sim \mathcal{N}(x, X - xx^T)$  and compute their projection  $v$  on

$$\{0, 1\}^{n \times (d+1)} \cap \left\{ x \in \mathbf{R}^{n \times (d+1)} : \sum_{j=0}^d x_{ij} = 1, i = 1, \dots, n \right\}.$$

For those sample schedules that meet the capacity constraints, we compute the total delay. Figure 2 shows the distribution of these objective values. We notice that the best delay found by randomization is equal



**Figure 2:** Histogram of randomized objective values (Total delay).

to 3, which matches the lower bound produced by solving problem (7). This shows that 3 is the globally optimum delay and that the corresponding randomized solution is optimal.

## Acknowledgements

The authors would like to acknowledge financial support from Eurocontrol grant C20083E/BM/05 and a gift from Google, Inc.

## References

- [Ali95] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5:13–51, 1995.
- [BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [CB76] E.G. Coffman and J.L. Bruno. *Computer and job-shop scheduling theory*. Wiley, 1976.
- [d'A05] A. d'Aspremont. Smooth optimization for sparse semidefinite programs. *ArXiv: math.OC/0512344*, 2005.
- [dEGJL05] A. d'Aspremont, L. El Ghaoui, M.I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *Advances in Neural Information Processing Systems*, 17:41–48, 2005.
- [FMOF01] E. Frazzoli, Z. H. Mao, J. H. Oh, and E. Feron. Aircraft conflict resolution via semi-definite programming. *AIAA J. of Guidance, Control, and Dynamics*, 24(1):79–86, 2001.
- [GW95] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, 1995.
- [HR00] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.
- [LS91] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.
- [NEG05] Arnab Nilim and Laurent El Ghaoui. Robust solutions to markov decision problems with uncertain transition matrices. *To appear in Operations Research*, 2005.
- [Nes04] Y. Nesterov. Smoothing technique and its application in semidefinite optimization. *CORE Discussion Paper No. 2004/73*, 2004.
- [PW95] S. Poljak and H. Wolkowicz. Convex relaxations of  $(0, 1)$ -quadratic programming. *Mathematics of Operations Research*, 20(3):550–561, 1995.
- [RSL77] D. J. Rosenkrantz, R.E. Stearns, and P.M. Lewis. An analysis of several heuristics for the traveling salesman problem. *SIAM J. Comput.*, 6:563–581, 1977.
- [Stu99] J. Sturm. Using SEDUMI 1.0x, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11:625–653, 1999.